Compositional Methods for Information-Hiding

Christelle Braun, Konstantinos Chatzikokolakis, Catuscia Palamidessi

Compositional Methods for Information-Hiding*

Christelle Braun Konstantinos Chatzikokolakis Catuscia Palamidessi
INRIA and LIX, École Polytechnique, Palaiseau, France
{braun,kostas,catuscia}@lix.polytechnique.fr

Abstract. Protocols for information-hiding often use randomized primitives to obfuscate the link between the observables and the information to be protected. The degree of protection provided by a protocol can be expressed in terms of the probability of error associated to the inference of the secret information.
We consider a probabilistic process calculus approach to the specification of such protocols, and we study how the operators affect the probability of error. In particular, we characterize constructs that have the property of not decreasing the degree of protection, and that can therefore be considered safe in the modular construction of protocols.
As a case study, we apply these techniques to the Dining Cryptographers, and we are able to derive a generalization of Chaum's strong anonymity result.

1 Introduction

During the last decade, internet activities have become an important part of many people's lives. As the number of these activities increases, there is a growing amount of personal information about the users that is stored in electronic form and that is usually transferred using public electronic means. This makes it feasible and often easy to collect, transfer and process a huge amount of information about a person. As a consequence, the need for mechanisms to protect the user's privacy is compelling.

We can categorize privacy properties based on the nature of the hidden information. *Data protection* usually refers to confidential data like the credit card number. *Anonymity*, on the other hand, concerns the identity of the user who performed a certain action. *Unlinkability* refers to the link between the information and the user, and *unobservability* regards the actions of a user.

Information-hiding protocols aim at ensuring a privacy property during an electronic transaction. For example, the voting protocol Foo 92 ([1]) allows a user to cast a vote without revealing the link between the voter and the vote. The anonymity protocol Crowds ([2]) allows a user to send a message on a public network without revealing the identity of the sender. These kinds of protocols often use *randomization* to introduce *noise*, thus limiting the inference power of a malicious observer.

1.1 Information theory

Recently it has been observed that at an abstract level information-hiding protocols can be viewed as *channels* in the information-theoretic sense. A channel consists of a set of

* This work has been partially supported by the INRIA DREI Équipe Associée PRINTEMPS and by the INRIA ARC project ProNoBiS.

input values $\mathcal{S}$, a set of output values $\mathcal{O}$ (the observables) and a transition matrix which gives the conditional probability $p(o|s)$ of producing o as the output when s is the input. In the case of privacy preserving protocols, $\mathcal{S}$ contains the secret information that we want to protect and $\mathcal{O}$ the facts that the attacker can observe. This framework allows us to apply concepts from information theory to reason about the knowledge that the attacker can gain about the input by observing the output of the protocol.

In the field of information flow and non-interference there have been various works [3–7] in which the *high information* and the *low information* are seen as the input and output respectively of a (noisy) channel. Non-interference is formalized in this setting as the converse of channel capacity.

Channel capacity has been also used in relation to anonymity in [8, 9]. These works propose a method to create covert communication by means of non-perfect anonymity.

A related line of work is [10, 11], where the main idea is to express the lack of (probabilistic) information in terms of entropy.

A different information-theoretic approach is taken in [12]. In this paper, the authors define as information leakage the difference between the a priori accuracy of the guess of the attacker, and the a posteriori one, after the attacker has made his observation. The accuracy of the guess is defined as the Kullback-Leibler distance between the *belief* (which is a weight attributed by the attacker to each input hypothesis) and the true distribution on the hypotheses.

1.2 Hypothesis testing

In information-hiding systems the attacker finds himself in the following scenario: he cannot directly detect the information of interest, namely the actual value of the random variable $S \in \mathcal{S}$, but he can discover the value of another random variable $O \in \mathcal{O}$ which depends on S according to a known conditional distribution. The attempt to infer S from O is called *hypothesis testing* (the "hypothesis" to be validated is the actual value of S), and it has been widely investigated in statistics. One of the most used approaches to this problem is the Bayesian method, which consists in assuming known the a priori probability distribution of the hypotheses, and deriving from that (and from the matrix of the conditional probabilities) the a posteriori distribution after a certain fact has been observed. It is well known that the best strategy for the adversary is to apply the MAP (Maximum Aposteriori Probability) criterion, which, as the name says, dictates that one should choose the hypothesis with the maximum a posteriori probability for the given observation. "Best" means that this strategy induces the smallest probability of error in the guess of the hypothesis. The probability of error, in this case, is also called *Bayes risk*. In [13], we proposed to define the *degree of protection* provided by a protocol as the Bayes risk associated to the matrix.

A major problem with the Bayesian method is that the a priori distribution is not always known. This is particularly true in security applications. In some cases, it may be possible to approximate the a priori distribution by statistical inference, but in most cases, especially when the input information changes over time, it may not. Thus other methods need to be considered, which do not depend on the a priori distribution. One such method is the one based on the so-called *Maximum Likelihood* criterion.

1.3 Contribution

In this paper we consider both the scenario in which the input distribution is known, in which case we consider the Bayes risk, and the one in which we have no information on the input distribution, or it changes over time. In this second scenario, we consider as degree of protection the probability of error associated to the Maximum Likelihood rule, averaged on all possible input distributions. It turns out that such average is equal to the value of the probability of error on the point of uniform distribution, which is much easier to compute.

Next, we consider a probabilistic process algebra for the specification of information-hiding protocols, and we investigate which constructs in the language can be used safely in the sense that by applying them to a term, the degree of protection provided by the term does not decrease. This provides a criterion to build specifications in a compositional way, while preserving the degree of protection.

We apply these compositional methods to the example of the Dining Cryptographers, and we are able to strengthen the strong anonymity result by Chaum. Namely we show that we can have strong anonymity even if some coins are unfair, provided that there is a spanning tree of fair ones. This result is obtained by adding processes representing coins to the specification and using the fact that this can be done with a safe construct.

The proofs are omitted for lack of space. They can be found in the report version of this paper, available on line at `http://www.lix.polytechnique.fr/~catuscia/papers/Anonymity/Compositional/report.pdf`.

1.4 Plan of the paper

In the next section we recall some basic notions. Section 3 introduces the language CCS$_p$. Section 4 shows how to model protocols and process terms as channels. Section 5 discusses hypothesis testing and presents some properties of the probability of error. Section 6 characterizes the constructs of CCS$_p$ which are safe. Section 7 applies previous results to find a new property of the Dining Cryptographers. Section 8 concludes.

2 Preliminaries

In this section we recall some basic notions of probability theory and probabilistic automata ([14, 15]).

A *discrete probability measure* over a set X is a function $\mu : 2^X \mapsto [0,1]$ such that $\mu(X) = 1$ and $\mu(\cup_i X_i) = \sum_i \mu(X_i)$ where X_i is a countable family of pairwise disjoint subsets of X. We denote the set of all discrete probability measures over X by $Disc(X)$. For $x \in X$, we denote by $\delta(x)$ (the *Dirac measure* on x) the probability measure that assigns probability 1 to $\{x\}$. If $\{c_i\}_i$ are convex coefficients, and $\{\mu_i\}_i$ are probability measures, we will denote by $\sum_i c_i\mu_i$ the probability measure defined as $(\sum_i c_i\mu_i)(Y) = \sum_i c_i\mu_i(Y)$.

A *probabilistic automaton* $\mathcal{M}$ is a tuple $(St, T_{init}, Act, \mathcal{T})$ where St is a set of states, $T_{init} \in St$ is the *initial state*, Act is a set of actions and $\mathcal{T} \subseteq St \times Act \times$

$Disc(St)$ is a *transition relation*. Intuitively, if $(T, a, \mu) \in \mathcal{T}$ then there is a transition from the state T performing the action a and leading to a distribution μ over the states of the automaton. (We use T for states because later in the paper states will be process terms, and S will be used for certain sequences of actions). We also write $T \xrightarrow{a} \mu$ if $(T, a, \mu) \in \mathcal{T}$. The idea is that the choice of transition among the available ones in $\mathcal{T}$ is performed nondeterministically, and the choice of the target state among the ones allowed by μ (i.e. those states T' such that $\mu(T') > 0$) is performed probabilistically. A probabilistic automaton $\mathcal{M}$ is *fully probabilistic* if from each state of $\mathcal{M}$ there is at most one transition available.

An *execution fragment* ϕ of a probabilistic automaton is a (possibly infinite) sequence $T_0 a_1 T_1 a_2 T_2 \ldots$ of alternating states and actions, such that for each i there is a transition $(T_i, a_{i+1}, \mu_i) \in \mathcal{T}$ and $\mu_i(T_{i+1}) > 0$. We will use $fst(\phi), lst(\phi)$ to denote the first and last state of a finite execution fragment ϕ respectively. An *execution* (or *history*) is an execution fragment such that $fst(\phi) = T_{init}$. An execution ϕ is maximal if it is infinite or there is no transition from $lst(\phi)$ in $\mathcal{T}$. We denote by $exec^*(\mathcal{M})$ the set of all the finite non-maximal executions of $\mathcal{M}$, and by $exec(\mathcal{M})$ the set of all the executions of $\mathcal{M}$.

A *scheduler* of a probabilistic automaton $\mathcal{M} = (St, T_{init}, Act, \mathcal{T})$ is a function

$$\zeta : exec^*(\mathcal{M}) \to \mathcal{T}$$

such that $\zeta(\phi) = (T, a, \mu) \in \mathcal{T}$ implies that $T = lst(\phi)$. The idea is that a scheduler selects a transition among the ones available in $\mathcal{T}$ and it can base its decision on the history of the execution. The *execution tree* of $\mathcal{M}$ relative to the scheduler ζ, denoted by $etree(\mathcal{M}, \zeta)$, is a fully probabilistic automaton $\mathcal{M}' = (St', T_{init}, Act, \mathcal{T}')$ such that $St' \subseteq exec(\mathcal{M})$, and $(\phi, a, \mu') \in \mathcal{T}'$ if and only if $\zeta(\phi) = (lst(\phi), a, \mu)$ for some μ, and $\mu'(\phi a T) = \mu(T)$. Intuitively, $etree(M, \zeta)$ is produced by unfolding the executions of $\mathcal{M}$ and resolving all nondeterministic choices using ζ. Note that $etree(\mathcal{M}, \zeta)$ is a fully probabilistic automaton.

Given a fully probabilistic automaton $\mathcal{M}$ we can define a probability space on the set $exec(\mathcal{M})$ of executions of $\mathcal{M}$ (see [14] for more details). Similarly, given a probabilistic automaton $\mathcal{M}$ and a scheduler ζ for $\mathcal{M}$, we can define a probability space on the set of traces of $\mathcal{M}$ by using the same construction on $etree(\mathcal{M}, \zeta)$, which is a fully probabilistic automaton.

3 CCS with internal probabilistic choice

We consider an extension of standard CCS ([16]) obtained by adding internal probabilistic choice. The resulting calculus CCS_p can be seen as a simplified version of the probabilistic π-calculus presented in [17, 18] and it is similar to the one considered in [19]. Like in those calculi, computations have both a probabilistic and a nondeterministic nature. The main conceptual novelty is a distinction between *observable* and *secret* actions, introduced for the purpose of specifying information-hiding protocols.

We assume a countable set Act of actions a, and we assume that it is partitioned into a set Sec of *secret actions* s, a set Obs of *observable actions* o, and the silent action τ. For each $s \in Sec$ we assume a complementary action $\overline{s} \in Sec$ such that $\overline{\overline{s}} = s$, and

$$\text{PROB}\ \frac{}{\sum\!\!\!\!\!\!\bigcirc_i\ p_i\,T_i \xrightarrow{\tau} \sum_i p_i\,\delta(T_i)} \qquad \text{ACT}\ \frac{j \in I}{\lfloor\!\!+\!\!\rfloor_I a_i.T_i \xrightarrow{a_j} \delta(T_j)}$$

$$\text{PAR1}\ \frac{T_1 \xrightarrow{a} \mu}{T_1 \mid T_2 \xrightarrow{a} \mu \mid T_2} \qquad \text{PAR2}\ \frac{T_2 \xrightarrow{a} \mu}{T_1 \mid T_2 \xrightarrow{a} T_1 \mid \mu} \qquad \text{REP}\ \frac{T \mid !T \xrightarrow{a} \mu}{!T \xrightarrow{a} \mu \mid !T}$$

$$\text{COM}\ \frac{T_1 \xrightarrow{a} \delta(T_1') \quad T_2 \xrightarrow{\overline{a}} \delta(T_2')}{T_1 \mid T_2 \xrightarrow{\tau} \delta(T_1' \mid T_2')} \qquad \text{RES}\ \frac{T \xrightarrow{b} \mu \qquad \phi \neq a, \overline{a}}{(\nu a)T \xrightarrow{b} (\nu a)\mu}$$

Table 1. The semantics of CCS$_p$.

the same for Obs. The silent action τ does not have a complementary action, so the notation $\overline{a}$ will imply that $a \in Sec$ or $a \in Obs$.

The syntax of CCS$_p$ is the following:

T	$::=$		*process term*
		$\sum\!\!\!\!\!\!\bigcirc_i\ p_i\,T_i$	*probabilistic choice*
	$\mid$	$\lfloor\!\!+\!\!\rfloor_i\ s_i.T_i$	*secret choice* $(s_i \in Sec)$
	$\mid$	$\lfloor\!\!+\!\!\rfloor_i\ r_i.T_i$	*nondeterministic choice* $(r_i \in Obs \cup \{\tau\})$
	$\mid$	$T \mid T$	*parallel composition*
	$\mid$	$(\nu a)\,T$	*restriction*
	$\mid$	$!T$	*replication*

All the summations in the syntax are finite. We will use the notation $T_1 \oplus_p T_2$ to represent a binary probabilistic choice $\sum\!\!\!\!\!\!\bigcirc_i\ p_i\,T_i$ with $p_1 = p$ and $p_2 = 1 - p$. Similarly we will use $a_1.T_1 \lfloor\!\!+\!\!\rfloor\, a_2.T_2$ to represent a binary secret or nondeterministic choice.

The semantics of a given CCS$_p$ term is a probabilistic automaton whose states are process terms, whose initial state is the given term, and whose transitions are those derivable from the rules in Table 1. We will use the notations (T, a, μ) and $T \xrightarrow{a} \mu$ interchangeably. We denote by $\mu \mid T$ the measure μ' such that $\mu'(T' \mid T) = \mu(T')$ for all processes T' and $\mu'(T'') = 0$ if T'' is not of the form $T' \mid T$, and similarly for $T \mid \mu$. Furthermore we denote by $(\nu a)\mu$ the measure μ' such that $\mu'((\nu a)T) = \mu(T)$, and $\mu'(T') = 0$ if T' is not of the form $(\nu a)T$.

Note that in the produced probabilistic automaton, all transitions to non-Dirac measures are silent. Note also that a probabilistic term generates exactly one (probabilistic) transition.

A transition of the form $T \xrightarrow{a} \delta(T')$, i.e. a transition having for target a Dirac measure, corresponds to a transition of a non-probabilistic automaton (a standard la-

beled transition system). Thus, all the rules of CCS$_p$ specialize to the ones of CCS except from PROB. The latter models the internal probabilistic choice: a silent τ transition is available from the sum to a measure containing all of its operands, with the corresponding probabilities.

A secret choice $\biguplus_i s_i.T_i$ produces the same transitions as the nondeterministic term $\biguplus_i r_i.T_i$, except for the labels.

The distinction between the two kind of labels influences the notion of scheduler for CCS$_p$: the secret actions are assumed to be *inputs* of the system, so a secret choice (with different guards) is determined by the input. The scheduler has to resolve only the residual nondeterminism.

In the following, we use the notation $X \rightharpoonup Y$ to represent the partial functions from X to Y, and $\phi_{|Sec}$ represents the projection of ϕ on Sec.

Definition 1. *Let T be a process in CCS$_p$ and $\mathcal{M}$ be the probabilistic automaton generated by T. A scheduler is a function $\zeta : Sec^* \to exec^* \rightharpoonup \mathcal{T}$ such that:*

(i) if $s = s_1 s_2 \dots s_n$ and $\phi_{|Sec} = s_1 s_2 \dots s_m$ with $m \leq n$, and
(ii) there exists a transition $(lst(\phi), a, \mu)$ such that, if $a \in Sec$ then $a = s_{m+1}$

then $\zeta(s)(\phi)$ is defined, and it is one of such transitions. We will write $\zeta_s(\phi)$ for $\zeta(s)(\phi)$.

Note that this definition of scheduler is different from the one used in probabilistic automaton, where the scheduler can decide to stop, even if a transition is allowed. Here the scheduler must proceed whenever a transition is allowed (provided that if it is labeled by a secret, that secret is the next one in the input string s).

We now adapt the definition of *execution tree* from the notion found in probabilistic automata. In our case, the execution tree depends not only on the scheduler, but also on the input.

Definition 2. *Let $\mathcal{M} = (St, T, Act, \mathcal{T})$ be the probabilistic automaton generated by a CCS$_p$ process T, where St is the set of processes reachable from T. Given an input s and a scheduler ζ, the* execution tree *of T for s and ζ, denoted by $etree(T, s, \zeta)$, is a fully probabilistic automaton $\mathcal{M}' = (St', T, Act, \mathcal{T}')$ such that $St' \subseteq exec(\mathcal{M})$, and $(\phi, a, \mu') \in \mathcal{T}'$ if and only if $\zeta_s(\phi) = (lst(\phi), a, \mu)$ for some μ, and $\mu'(\phi a T) = \mu(T)$.*

4 Modeling protocols for information-hiding

We propose here an abstract model for information-hiding protocols, and we show how to represent this model in CCS$_p$. An extended example is presented in Section 7.

4.1 Protocols as channels

We view protocols as *channels* in the information-theoretic sense [20]. The secret information that the protocol is trying to conceal constitutes the input of the channel, and the observables constitute the outputs. The set of the possible inputs and that of the possible outputs will be denoted by $\mathcal{S}$ and $\mathcal{O}$ respectively. We assume that $\mathcal{S}$ and $\mathcal{O}$ are

of finite cardinality m and n respectively. We also assume a discrete probability distribution over the inputs, which we will denote by $\vec{\pi} = (\pi_{s_1}, \pi_{s_2}, \ldots, \pi_{s_m})$, where π_s is the probability of the input s.

To fit the model of the channel, we assume that at each run, the protocol is given exactly one secret s_i to conceal. This is not a restriction, because the s_i's can be complex information like sequences of keys or tuples of individual data. During the run, the protocol may use randomized operations to increase the level of uncertainty about the secrets and obfuscate the link with the observables. It may also have internal interactions between internal components, or other forms of nondeterministic behavior, but let us rule out this possibility for the moment, and consider a purely probabilistic protocol. We also assume there is exactly one output from each run of the protocol, and again, this is not a restrictive assumption because the elements of $\mathcal{O}$ can be structured data.

Given an input s, a run of the protocol will produce each $o \in \mathcal{O}$ with a certain probability $p(o|s)$ which depends on s and on the randomized operations performed by the protocol. Note that $p(o|s)$ depends only on the probability distributions on the mechanisms of the protocol, and not on the input distribution. The probabilities $p(o|s)$, for $s \in \mathcal{S}$ and $o \in \mathcal{O}$, constitute a $m \times n$ array M which is called the *matrix* of the channel, where the rows are indexed by the elements of $\mathcal{S}$ and the columns are indexed by the elements of $\mathcal{O}$. We will use the notation $(\mathcal{S}, \mathcal{O}, M)$ to represent the channel.

Note that the input distribution $\vec{\pi}$ and the probabilities $p(o|s)$ determine a distribution on the output. We will represent by $p(o)$ the probability of $o \in \mathcal{O}$. Thus both the input and the output can be considered *random variables*. We will denote these random variables by S and O.

If the protocol contains some forms of nondeterminism, like internal components giving rise to different interleaving and interactions, then the behavior of the protocol, and in particular the output, will depend on the scheduling policy. We can reduce this case to previous (purely probabilistic) scenario by assuming a scheduler ζ which resolves the nondeterminism entirely. Of course, the conditional probabilities, and therefore the matrix, will depend on ζ, too. We will express this dependency by using the notation M_ζ.

4.2 Process terms as channels

A given CCS$_p$ term T can be regarded as a protocol in which the input is constituted by sequences of secret actions, and the output by sequences of observable actions. We assume that only a finite set of such sequences is relevant. This is certainly true if the term is terminating, which is usually the case in security protocols where each session is supposed to terminate in finite time.

Thus the set S could be, for example, the set of all sequences of secret actions up to a certain length (for example, the maximal length of executions) and analogously O could be the set of all sequences of observable actions up to a certain length. To be more general, we will just assume $\mathcal{S} \subseteq_{fin} Sec^*$ and $\mathcal{O} \subseteq_{fin} Obs^*$.

Definition 3. *Given a term T and a scheduler $\zeta : \mathcal{S} \to exec^* \to \mathcal{T}$, the matrix $M_\zeta(T)$ associated to T under ζ is defined as the matrix such that, for each $s \in \mathcal{S}$ and $o \in \mathcal{O}$,*

$p(o|s)$ is the probability of the set of the maximal executions in $etree(T, s, \zeta)$ whose projection in Obs is o.

5 Inferring the secrets from the observables

In this section we discuss possible methods by which an adversary can try to infer the secrets from the observables, and consider the corresponding probability of error, that is, the probability that the adversary draws the wrong conclusion. We regard the probability of error as a representative of the degree of protection provided by the protocol, and we study its properties with respect to the associated matrix.

We start by defining the notion of *decision function*, which represents the guess the adversary makes about the secrets, for each observable. This is a well-known concept, particularly in the field of *hypothesis testing*, where the purpose is to try to discover the valid hypothesis from the observed facts, knowing the probabilistic relation between the possible hypotheses and their consequences. In our scenario, the hypotheses are the secrets.

Definition 4. *A decision function for a channel $(\mathcal{S}, \mathcal{O}, M)$ is any function $f : \mathcal{O} \to \mathcal{S}$.*

Given a channel $(\mathcal{S}, \mathcal{O}, M)$, an input distribution $\vec{\pi}$, and a decision function f , the *probability of error* $\mathcal{P}(f, M, \vec{\pi})$ is the average probability of guessing the wrong hypothesis by using f, weighted on the probability of the observable (see for instance [20]). The probability that, given o, s is the wrong hypothesis is $1-p(s|o)$ (with a slight abuse of notation, we use $p(\cdot|\cdot)$ to represent also the probability of the input given the output). Hence we have:

Definition 5 (Probability of error, [20]). $\mathcal{P}(f, M, \vec{\pi}) \;=\; 1 - \sum_{\mathcal{O}} p(o)p(f(o)|o)$

Given a channel $(\mathcal{S}, \mathcal{O}, M)$, the best decision function that the adversary can use, namely the one that minimizes the probability of error, is the one associated to the so-called MAP rule, which prescribes choosing the hypothesis s which has *Maximum Aposteriori Probability* (for a given $o \in \mathcal{O}$), namely the s for which $p(s|o)$ is maximum. The fact that the MAP rule represent the 'best bet' of the adversary is rather intuitive, and well known in the literature. We refer to [20] for a formal proof.

The MAP rule is used in the so-called *Bayesian approach* to hypothesis testing, and the corresponding probability of error is also known as *Bayes risk*. We will denote it by $\mathcal{P}_{MAP}(M, \vec{\pi})$. The following characterization is an immediate consequence of Definition 5 and of the Bayes theorem $p(s|o) = p(o|s)\pi_s/p(o)$.

$$\mathcal{P}_{MAP}(M, \vec{\pi}) \;=\; 1 - \sum_{\mathcal{O}} \max_s (p(o|s)\pi_s)$$

It is natural then to define the degree of protection associated to a process term as the infimum probability of error that we can obtain from this term under every compatible scheduler (in a given class).

In the following, we assume the class of schedulers $\mathcal{A}$ to be the set of all the schedulers compatible with the given input $\mathcal{S}$.

It turns out that the infimum probability of error on $\mathcal{A}$ is actually a minimum:

Proposition 1. *For every CCS$_p$ process T we have*

$$\inf_{\zeta \in \mathcal{A}} \mathcal{P}_{MAP}(M_\zeta(T), \vec{\pi}) \;=\; \min_{\zeta \in \mathcal{A}} \mathcal{P}_{MAP}(M_\zeta(T), \vec{\pi})$$

Thanks to previous proposition, we can define the degree of protection provided by a protocols in terms of the minimum probability of error.

Definition 6. *Given a CCS$_p$ process T, the protection $Pt_{MAP}(T)$ provided by T, in the Bayesian approach, is given by*

$$Pt_{MAP}(T, \vec{\pi}) \;=\; \min_{\zeta \in \mathcal{A}} \mathcal{P}_{MAP}(M_\zeta(T), \vec{\pi})$$

The problem with the MAP rule is that it assumes that the input distribution is known to the adversary. This is often not the case, so it is natural to try to approximate it with some other rule. One such rule is the so-called ML rule, which prescribes the choice of the s which has *Maximum Likelihood* (for a given $o \in \mathcal{O}$), namely the s for which $p(o|s)$ is maximum. The name comes from the fact that $p(o|s)$ is called the *likelihood* of s given o. We will denote the corresponding probability of error by $\mathcal{P}_{ML}(M, \vec{\pi})$. The following characterization is an immediate consequence of Definition 5 and of the Bayes Theorem.

$$\mathcal{P}_{ML}(M, \vec{\pi}) \;=\; 1 - \sum_{\mathcal{O}} \max_s (p(o|s))\pi_s$$

It has been shown (see for instance [21]) that under certain conditions on the matrix, the ML rule approximates indeed the MAP rule, in the sense that by repeating the protocol the adversary can make the probability of error arbitrarily close to 0, with either rule.

We could now define the degree of protection provided by a term T under the ML rule as the minimum $\mathcal{P}_{ML}(M_\zeta(T), \vec{\pi})$, but it does not seem reasonable to give a definition that depends on the input distribution, since the main reason to apply a non-Bayesian approach is that indeed we do not know the input distribution. Instead, we define the degree of protection associated to a process term as the *average* probability of error with respect to all possible distributions $\vec{\pi}$:

Definition 7. *Given a CCS$_p$ process T, the protection $Pt_{ML}(T)$ provided by T, in the Maximum Likelihood approach, is given by*

$$Pt_{ML}(T) \;=\; \min_{\zeta \in \mathcal{A}} \; (m-1)! \int_{\vec{\pi}} \mathcal{P}_{ML}(M_\zeta(T), \vec{\pi}) \; d\vec{\pi}$$

In the above definition, $(m-1)!$ represents a normalization function: $\frac{1}{(m-1)!}$ is the hyper-volume of the domain of all possible distributions $\vec{\pi}$ on $\mathcal{S}$, namely the $(m-1)$-dimensional space of points $\vec{\pi}$ such that $0 \leq \pi_s \leq 1$ and $0 \leq \sum_{s \in S} \pi_s = 1$ (where m is the cardinality of $\mathcal{S}$).

Fortunately, it turns out that this definition is equivalent to a much simpler one: the average value of the probability of error, under the Maximum Likelihood rule, can be obtained simply by computing $\mathcal{P}_{ML}$ on the uniform distribution $\vec{\pi}_u = (\frac{1}{m}, \frac{1}{m}, \ldots, \frac{1}{m})$.

Theorem 1.

$$Pt_{ML}(T) \;=\; \min_{\zeta\in\mathcal{A}}\; \mathcal{P}_{ML}(M_\zeta(T),\vec{\pi}_u)$$

The next corollary follows immediately from Theorem 1 and from the definitions of $\mathcal{P}_{MAP}$ and $\mathcal{P}_{ML}$.

Corollary 1.

$$Pt_{ML}(T) \;=\; \min_{\zeta\in\mathcal{A}}\; \mathcal{P}_{MAP}(M_\zeta(T),\vec{\pi}_u)$$

We conclude this section with some properties of $\mathcal{P}_{MAP}$. Note that the same properties hold also for $\mathcal{P}_{ML}$ on the uniform distribution, because $\mathcal{P}_{ML}(M,\vec{\pi}_u) = \mathcal{P}_{MAP}(M,\vec{\pi}_u)$.

The next proposition shows that the probabilities of error are *concave* functions with respect to the space of matrices.

Proposition 2. *Consider a family of channels $\{(\mathcal{S},\mathcal{O},M_i)\}_{i\in I}$, and a family $\{c_i\}_{i\in I}$ of convex coefficients, namely $0\le c_i\le 1$ for all $i\in I$, and $\sum_{i\in I} c_i \;=\; 1$. Then:*

$$\mathcal{P}_{MAP}(\sum_{i\in I} c_i\, M_i,\vec{\pi}) \;\ge\; \sum_{i\in I} c_i\, \mathcal{P}_{MAP}(M_i,\vec{\pi})$$

Corollary 2. *Consider a family of channels $\{(\mathcal{S},\mathcal{O},M_i)\}_{i\in I}$, and a family $\{c_i\}_{i\in I}$ of convex coefficients. Then:*

$$\mathcal{P}_{MAP}(\sum_{i\in I} c_i\, M_i,\vec{\pi}) \;\ge\; \min_{i\in I}\; \mathcal{P}_{MAP}(M_i,\vec{\pi})$$

The next proposition shows that if we transform the observables, and collapse the columns corresponding to observables which have become the same after the transformation, the probability of error does not decrease.

Proposition 3. *Consider a channel $(\mathcal{S},\mathcal{O},M)$, where M has conditional probabilities $p(o|s)$, and a transformation of the observables $f:\mathcal{O}\to\mathcal{O}'$. Let M' be the matrix whose conditional probabilities are $p'(o'|s)=\sum_{f(o)=o'} p(o|s)$ and consider the new channel $(\mathcal{S},\mathcal{O}',M')$. Then:*

$$\mathcal{P}_{MAP}(M',\vec{\pi}) \;\ge\; \mathcal{P}_{MAP}(M,\vec{\pi})$$

The following propositions are from the literature.

Proposition 4 ([21]). *Given $\mathcal{S}$, $\mathcal{O}$, let M be a matrix indexed on $\mathcal{S}$, $\mathcal{O}$ such that all the rows of M are equal, namely $p(o|s)=p(o|s')$ for all $o\in\mathcal{O}$, $s,s'\in\mathcal{S}$. Then,*

$$\mathcal{P}_{MAP}(M,\vec{\pi}) \;=\; 1-\max_{s}\; \pi_s$$

Furthermore $\mathcal{P}_{MAP}(M,\vec{\pi})$ is the maximum probability of error, i.e. for every other matrix M' indexed on $\mathcal{S}$, $\mathcal{O}$ we have:

$$\mathcal{P}_{MAP}(M,\vec{\pi}) \;\ge\; \mathcal{P}_{MAP}(M',\vec{\pi})$$

Proposition 5 ([22]). *Given a channel $(\mathcal{S}, \mathcal{O}, M)$, the rows of M are equal (and hence the probability of error is maximum) if and only if $p(s|o) = \pi_s$ for all $s \in \mathcal{S}$, $o \in \mathcal{O}$.*

The condition $p(s|o) = \pi_s$ means that the observation does not give any additional information concerning the hypothesis. In other words, the *a posteriori* probability of s coincides with its *a priori* probability. The property $p(s|o) = \pi_s$ for all $s \in \mathcal{S}$ and $o \in \mathcal{O}$ was used as a definition of (strong) anonymity by Chaum [23] and was called *conditional anonymity* by Halpern and O'Neill [24].

6 Safe constructs

In this section we investigate constructs of the language CCS_p which are *safe* with respect to the protection of the secrets.

We start by giving some conditions that will allow us to ensure the safety of the parallel and the restriction operators.

Definition 8. *Consider process terms T_1, T_2, and observables $o_1, o_2, \ldots, o_k$ such that*

(i) T_1 does not contain any secret action, and
(ii) the observable actions of T_1 are included in $o_1, o_2, \ldots, o_k$.

Then we say that T_1 and $o_1, o_2, \ldots, o_k$ are safe with respect to T_2.

The following theorem states our main results for Pt_{MAP}. Note that they are also valid for Pt_{ML}, because $Pt_{ML}(T) = Pt_{MAP}(T, \vec{\pi}_u)$.

Theorem 2. *The probabilistic choice, the nondeterministic choice, and a restricted form of parallel composition are safe constructs, namely, for every input probability π, and any terms $T_1, T_2, \ldots, T_h$, we have*

$$(1) \qquad Pt_{MAP}(\sum_i p_i\, T_i, \vec{\pi}) \;\geq\; \sum_i p_i\, Pt_{MAP}(T_i, \vec{\pi}) \;\geq\; \min_i\, Pt_{MAP}(T_i, \vec{\pi})$$

$$(2) \qquad Pt_{MAP}(\biguplus_i o_i.T_i, \vec{\pi}) \;=\; \min_i Pt_{MAP}(T_i, \vec{\pi})$$

$$(3) \quad Pt_{MAP}((\nu o_1, o_2, \ldots, o_k)\,(T_1 \mid T_2)) \;\geq\; Pt_{MAP}(T_2, \vec{\pi})$$

if T_1 and $o_1, o_2, \ldots, o_k$ are safe w.r.t. T_2

Unfortunately the safety property does not hold for the secret choice. The following is a counterexample.

Example 1. Let $Sec = \{s_1, s_2\}$ and assume that $\mathcal{S}$ does not contain the empty sequence. Let $T = o_1.0 \uplus o_2.0$. Then $Pt_{MAP}(T, \vec{\pi})$ is maximum (i.e. $Pt_{MAP}(T, \vec{\pi}) = 1 - \max \vec{\pi}$) because for every sequence $s \in \mathcal{S}$ we have $p(o_1|s) = p(o_2|s)$. Let $T' = s_1.T \uplus s_2.T$. We can now define a scheduler such that, if the secret starts with s_1, it selects o_1, and if the secret starts with s_2, it selects o_2. Hence, under this scheduler, $p(o_1|s_1 s) = p(o_2|s_2 s) = 1$ while $p(o_1|s_2 s) = p(o_2|s_1 s) = 0$. Therefore $Pt_{MAP}(T', \vec{\pi}) = 1 - p_1 - p_2$ where p_1 and p_2 are the maximum probabilities of the

secrets of the form $s_1 s$ and $s_2 s$, respectively. Note now that either $\max \vec{\pi} = p_1$ or $\max \vec{\pi} = p_2$ because of the assumption that $\mathcal{S}$ does not contain the empty sequence. Let $\vec{\pi}$ be such that both p_1 and p_2 are positive. Then $1 - p_1 - p_2 < 1 - \max \vec{\pi}$, hence $Pt_{MAP}(T', \vec{\pi}) < Pt_{MAP}(T, \vec{\pi})$.

The reason why we need the condition *(i)* in Definition 8 for the parallel operator is analogous to the case of secret choice. The following is a counterexample.

Example 2. Let Sec and $\mathcal{S}$ be as in Example 1. Define $T_1 = s_1.0 \boxplus s_2.0$ and $T_2 = o_1.0 \boxplus o_2.0$. Clearly, $Pt_{MAP}(T_2, \vec{\pi}) = 1 - \max \vec{\pi}$. Consider now the term $T_1 \mid T_2$ and define a scheduler that first executes an action s in T_1 and then, if s is s_1, it selects o_1, while if s is s_2, it selects o_2. The rest proceeds like in Example 1, where $T' = T_1 \mid T_2$ and $T = T2$.

The reason why we need the condition *(ii)* in Definition 8 is that without it the parallel operator may create different interleavings, thus increasing the possibility of an adversary discovering the secrets. The following is a counterexample.

Example 3. Let Sec and $\mathcal{S}$ be as in Example 1. Define $T_1 = o.0$ and $T_2 = s_1.(o_1.0 \oplus_{.5} o_2.0) \boxplus s_2.(o_1.0 \oplus_{.5} o_2.0)$. It is easy to see that $Pt_{MAP}(T_2, \vec{\pi}) = 1 - \max \vec{\pi}$. Consider the term $T_1 \mid T_2$ and define a scheduler that first executes an action s in T_2 and then, if s is s_1, it selects first T_1 and then the continuation of T_2, while if s is s_2, it selects first the continuation of T_2 and then T_1. Hence, under this scheduler, $p(oo_1|s_1 s) = p(oo_2|s_1 s) = .5$ and also $p(o_1 o|s_2 s) = p(o_2 o|s_2 s) = .5$ while $p(oo_1|s_2 s) = p(oo_2|s_2 s) = 0$ and $p(o_1 o|s_1 s) = p(o_2 o|s_1 s) = 0$. Therefore we have that $Pt_{MAP}(T, \vec{\pi}) = 1 - p_1 - p_2$ where p_1 and p_2 are the maximum probabilities of the secrets of the form $s_1 s$ and $s_2 s$, respectively. Following the same reasoning as in example 1, we have that for certain $\vec{\pi}$, $Pt_{MAP}(T', \vec{\pi}) < Pt_{MAP}(T, \vec{\pi})$.

7 A case study: the Dining Cryptographers

In this section we consider the Dining Cryptographers (DC) protocol proposed by Chaum in [23], we show how to describe it in CCS$_p$, and we apply the results of previous section to obtain a generalization of Chaum's strong anonymity result.

In its most general formulation, the DC consists of a multigraph where one of the nodes (cryptographers) may be secretly designated to pay for the dinner. The cryptographers would like to find out whether there is a payer or not, but without either discovering the identity of the payer, nor revealing it to an external observer. The problem can be solved as follows: we put on each edge a probabilistic coin, which can give either 0 or 1. The coins get tossed, and each cryptographer computes the binary sum of all (the results of) the adjacent coins. Furthermore, it adds 1 if it is designated to be the payer. Finally, all the cryptographers declare their result.

It is easy to see that this protocol solves the problem of figuring out the existence of a payer: the binary sum of all declarations is 1 if and only if there is a payer, because all the coins get counted twice, so their contribution to the total sum is 0.

The property we are interested in, however, is the anonymity of the system. Chaum proved that the DC is strongly anonymous if all the coins are fair, i.e. they give 0 and 1

$$
\begin{aligned}
Crypt_i &= c_{i,i_1}(x_1) \,.\, \ldots \,.\, c_{i,i_k}(x_k) \,.\, pay_i(x) \,.\, \bar{d}_i\langle x_1 + \ldots + x_k + x\rangle \\
Coin_h &= \bar{c}_{\ell,h}\langle 0\rangle \,.\, \bar{c}_{r,h}\langle 0\rangle.0 \;\oplus_{p_h}\; \bar{c}_{\ell,h}\langle 1\rangle \,.\, \bar{c}_{r,h}\langle 1\rangle.0 \\
Collect &= d_1(y_1) \,.\, d_2(y_2) \,.\, \ldots \,.\, d_n(y_n) \,.\, \overline{out}\langle y_1, y_2, \ldots, y_n\rangle \\
DC &= (\nu \vec{c})(\nu \vec{d})(\textstyle\prod_i Crypt_i \mid \prod_h Coin_h \mid Collect)
\end{aligned}
$$

Table 2. The dining cryptographers protocol expressed in CCS_p.

with equal probability, and the multigraph is connected, namely there is a path between each pair of nodes. To state formally the property, let us denote by s the secret identity of the payer, and by o the collection of the declarations of the cryptographers.

Theorem 3 ([23]). *If the multigraph is connected, and the coins are fair, then DC is strongly anonymous, namely for every s and o, $p(s|o) = \pi_s$ holds.*

We are now going to show how to express the DC in CCS_p. We start by introducing a notation for value-passing in CCS_p, following standard lines.

$$
\begin{aligned}
Input \quad & c(x).T = \lfloor + \rfloor_v \, c_v.T[v/x] \\
Output \quad & \bar{c}\langle v\rangle = \bar{c}_v
\end{aligned}
$$

The protocol can be described as the parallel composition of the cryptographers processes $Crypt_i$, of the coin processes $Coin_h$, and of a process *Collect* whose purpose is to collect all the declarations of the cryptographers, and output them in the form of a tuple. See Table 2. In this protocol, the secret actions are pay_i. All the others are observable actions.

Each coin communicates with two cryptographers. $c_{i,h}$ represents the communication channel between $Coin_h$ and $Crypt_i$ if h is indeed the index of a coin, otherwise it represents a communication channel "with the environment". We will call the latter *external*. In the original definition of the DC there are no external channels, we have added them to prove a generalization of Chaum's result. They could be interpreted as a way for the environment to influence the computation of the cryptographers and hence test the system, for the purpose of discovering the secret.

We are now ready to state our generalization of Chaum's result.

Theorem 4. *A DC is strongly anonymous if it has a spanning tree consisting of fair coins only.*

Proof. Consider the term DC in Table 2. Remove all the coins that do not belong to the spanning tree, and the corresponding restriction operators. Let T be the process term obtained this way. Let $\mathcal{A}$ be the class of schedulers which select the value 0 for all the external channels. This situation corresponds to the original formulation of Chaum and so we can apply Chaum's result (Theorem 3) and Proposition 5 to conclude that all the rows of the matrix M are the same and hence, by Proposition 4, $\mathcal{P}_{MAP}(M, \vec{\pi}) = 1 - \max_i \pi_i$.

Consider now one of the removed coins, h, and assume, without loss of generality, that $c_{\ell,h}(x)$, $c_{r,h}(x)$ are the first actions in the definitions of $Crypt_\ell$ and $Crypt_r$. Consider the class of schedulers $\mathcal{B}$ that selects value 1 for x in these actions. The matrix M' that we obtain is isomorphic to M: the only difference is that each column o is now mapped to a column $o + w$, where w is a tuple that has 1 in the ℓ and r positions, and 0 in all other positions, and $+$ represents the componentwise binary sum. Since this map is a bijection, we can apply Proposition 3 in both directions and derive that $\mathcal{P}_{MAP}(M', \vec{\pi}) = 1 - \max_i \pi_i$.

We can conclude, therefore, that $Pt_{MAP}(T, \vec{\pi}) = 1 - \max_i \pi_i$ in the class of schedulers $\mathcal{A} \cup \mathcal{B}$.

By repeating the same reasoning on each of the removed coins, we can conclude that $Pt_{MAP}(T, \vec{\pi}) = 1 - \max_i \pi_i$ for any scheduler ζ of T.

Consider now the term $T' = (\nu c_{\ell,h} c_{r,h})(Coin_h \mid T)$ obtained from T by adding back the coin h. By applying Theorem 2 we can deduce that $Pt_{MAP}(T', \vec{\pi}) \geq Pt_{MAP}(T, \vec{\pi})$. By repeating this reasoning, we can add back all the coins, one by one, and obtain the original DC. Hence we can conclude that $Pt_{MAP}(DC, \vec{\pi}) \geq Pt_{MAP}(T, \vec{\pi}) = 1 - \max_i \pi_i$ and, since $Pt_{MAP}(T, \vec{\pi})$ is maximum, we have $Pt_{MAP}(DC, \vec{\pi}) = 1 - \max_i \pi_i$, which concludes the proof.

Interestingly, also the other direction of Theorem 4 holds. We report this result for completeness, however we have proved it by using traditional methods, not by applying the compositional methods of Section 6.

Theorem 5. *A DC is strongly anonymous only if it has a spanning tree consisting of fair coins only.*

8 Conclusion and future work

In this paper we have investigated the properties of the probability of error associated to a given information-hiding protocol, and we have investigated CCS$_p$constructs that are safe, i.e. that are guaranteed not to decrease the protection of the protocol. Then we have applied these results to strengthen a result of Chaum: the dining cryptographers are strongly anonymous if and only if they have a spanning tree of fair coins.

In the future, we would like to extend our results to other constructs of the language. This is not possible in the present setting, as the examples after Theorem 2 show. The problem is related to the scheduler: the standard notion of scheduler is too powerful and can leak secrets, by depending on the secret choices that have been made in the past. All the examples after Theorem 2 are based on this kind of problem. In [25], we have studied the problem and we came out with a language-based solution to restrict the power of the scheduler. We are planning to investigate whether such approach could be exploited here to guarantee the safety of more constructs.

References

1. Fujioka, A., Okamoto, T., Ohta, K.: A practical secret voting scheme for large scale elections. In: ASIACRYPT '92, Springer-Verlag (1993) 244–251

2. Reiter, M.K., Rubin, A.D.: Crowds: anonymity for Web transactions. ACM Transactions on Information and System Security **1** (1998) 66–92
3. McLean, J.: Security models and information flow. In: Proc. of SSP, IEEE (1990) 180–189
4. Gray, III, J.W.: Toward a mathematical foundation for information flow security. In: Proc. of SSP '91, IEEE (1991) 21–35
5. Clark, D., Hunt, S., Malacaria, P.: Quantitative analysis of the leakage of confidential data. In: Proc. of QAPL 2001. ENTCS, Elsevier Science B.V. (2001) 238–251
6. Clark, D., Hunt, S., Malacaria, P.: Quantified interference for a while language. In: Proc. of QAPL 2004. Volume 112 of ENTCS., Elsevier Science B.V. (2005) 149–166
7. Lowe, G.: Quantifying information flow. In: Proc. of CSFW 2002, IEEE Computer Society Press (2002) 18–31
8. Moskowitz, I.S., Newman, R.E., Crepeau, D.P., Miller, A.R.: Covert channels and anonymizing networks. In Jajodia, S., Samarati, P., Syverson, P.F., eds.: WPES, ACM (2003) 79–88
9. Moskowitz, I.S., Newman, R.E., Syverson, P.F.: Quasi-anonymous channels. In: IASTED CNIS. (2003) 126–131
10. Serjantov, A., Danezis, G.: Towards an information theoretic metric for anonymity. In: Proc. of PET. Volume 2482 of LNCS., Springer (2002) 41–53
11. Díaz, C., Seys, S., Claessens, J., Preneel, B.: Towards measuring anonymity. In: Proc. of PET. Volume 2482 of LNCS., Springer (2002) 54–68
12. Clarkson, M.R., Myers, A.C., Schneider, F.B.: Belief in information flow. Journal of Computer Security (2008) To appear.
13. Chatzikokolakis, K., Palamidessi, C., Panangaden, P.: Probability of error in information-hiding protocols. In: Proc. of CSF, IEEE (2007) 341–354
14. Segala, R.: Modeling and Verification of Randomized Distributed Real-Time Systems. PhD thesis, MIT (1995) Tech. Rep. MIT/LCS/TR-676.
15. Segala, R., Lynch, N.: Probabilistic simulations for probabilistic processes. Nordic Journal of Computing **2** (1995) 250–273
16. Milner, R.: Communication and Concurrency. International Series in Computer Science. Prentice Hall (1989)
17. Herescu, O.M., Palamidessi, C.: Probabilistic asynchronous π-calculus. In: Proceedings of FOSSACS. Volume 1784 of LNCS., Springer (2000) 146–160
18. Palamidessi, C., Herescu, O.M.: A randomized encoding of the π-calculus with mixed choice. Theoretical Computer Science **335** (2005) 373–404
19. Deng, Y., Palamidessi, C., Pang, J.: Compositional reasoning for probabilistic finite-state behaviors. In: Processes, Terms and Cycles: Steps on the Road to Infinity. Volume 3838 of LNCS. Springer (2005) 309–337
20. Cover, T.M., Thomas, J.A.: Elements of Information Theory. John Wiley & Sons, Inc. (1991)
21. Chatzikokolakis, K., Palamidessi, C., Panangaden, P.: Anonymity protocols as noisy channels. Information and Computation (2007) To appear.
22. Bhargava, M., Palamidessi, C.: Probabilistic anonymity. In: Proceedings of CONCUR. Volume 3653 of LNCS., Springer (2005) 171–185
23. Chaum, D.: The dining cryptographers problem: Unconditional sender and recipient untraceability. Journal of Cryptology **1** (1988) 65–75
24. Halpern, J.Y., O'Neill, K.R.: Anonymity and information hiding in multiagent systems. Journal of Computer Security **13** (2005) 483–512
25. Chatzikokolakis, K., Palamidessi, C.: Making random choices invisible to the scheduler. In: Proc. of CONCUR'07. Volume 4703 of LNCS., Springer (2007) 42–58